



Interfacing Matlab/Simulink with V-REP for an Easy Development of Sensor-Based Control Algorithms for Robotic Platforms

Riccardo Spica, Giovanni Claudio, Fabien Spindler, Paolo Robuffo Giordano

► To cite this version:

Riccardo Spica, Giovanni Claudio, Fabien Spindler, Paolo Robuffo Giordano. Interfacing Matlab/Simulink with V-REP for an Easy Development of Sensor-Based Control Algorithms for Robotic Platforms. 2014. hal-01332960

HAL Id: hal-01332960

<https://hal.inria.fr/hal-01332960>

Submitted on 17 Jun 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Interfacing Matlab/Simulink with V-REP for an Easy Development of Sensor-Based Control Algorithms for Robotic Platforms

Riccardo Spica, Giovanni Claudio, Fabien Spindler and Paolo Robuffo Giordano

I. OUTLINE

In this presentation we will show how to interface Matlab/Simulink with the open-source 3D simulation engine V-REP¹ exploiting the popular ROS² communication middleware.

The Matlab/Simulink environment is a very convenient possibility for developing, debugging and testing complex control algorithms in a “fast prototyping” fashion. One can code algorithms either in the matlab scripting language or by including external C/C++ code using, e.g., s-functions. Also, the automatic code generation feature makes it possible to speed up the execution time and even to deploy the generated binaries to other platforms (e.g., the robot itself). Finally, the use of scopes, displays, and the other visualization and matlab post-processing tools represents a great added value during the debugging and testing phase of the algorithms.

On the other hand, to the best of our knowledge, Matlab lacks an easy-to-use 3D physical simulation engine specifically tailored to *robotics applications*. In this respect, V-REP represents a valid alternative: it consists of an open-source state-of-the-art (and *freely available* for academic use) 3D physical simulation engine which is becoming more and more widespread in the robotics community thanks to its flexibility (possibility to simulate many different robotic platforms), dynamical engine (it supports ODE, Bullet and Vortex), and finally customizability (it offers many different possibilities to include one’s own code or to interface it with the external world). Concerning this latter point, V-REP comes with a native support of ROS that makes it easy to exchange data with other modules, e.g., for sending the robot state and sensor data to an external planner/controller and receiving back the motion commands.

This presentation will then focus on how to interface the matlab/simulink environment with V-REP using the ROS communication libraries (the publisher/subscriber paradigm) for fast prototyping of robot control algorithms. We will

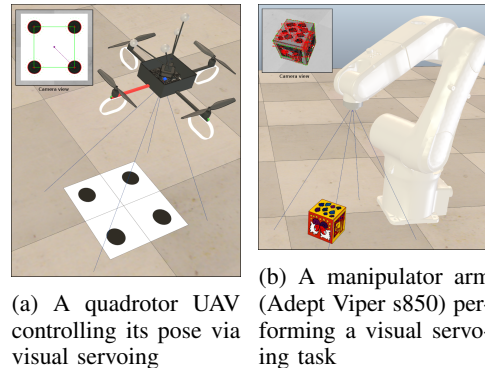


Fig. 1: Two screenshots of the V-REP Simulation environment

first show how to embed ROS nodes in simulink by including custom C S-Functions representing ROS topics to be listened/published. This will make it possible for simulink to exchange data with V-REP in real-time for obtaining the robot data and computing the needed control actions. Then, we will demonstrate our architecture in two simulated scenarios: (i) visual control of a quadrotor UAV and (ii) visual control of an industrial manipulator. The first scenario will involve a quadrotor UAV equipped with a IMU and a down-looking camera meant to control its pose w.r.t. a ground target by means of a visual servoing law [1]. The second scenario will consider the same situation for a fixed manipulator with an eye-in-hand camera performing a classical visual servoing task [2]. Figure 1 shows two snapshots of the simulation, while a video is available online at <http://www.irisa.fr/lagadic/demo/demo-vrep/demo-vrep.html>.

In both cases, the (simulated) camera images generated in V-REP are first pre-processed in an independent ROS node by using the well-established ViSP library³ [3] for extracting the salient features of the tracked object. The location of the features, together with the quadrotor/manipulator states, are then sent via ROS to simulink which implements the servoing loop (and the flying controller in the quadrotor case). The motion commands (thrust/torques or joint velocities) are then sent back to the V-REP simulation. The overall software architecture is shown in Fig. 2. The time synchronisation between matlab/simulink and V-REP is achieved by relying on a *soft real-time* execution: both V-REP (natively) and matlab/simulink (by means of a custom s-function) are forced

R. Spica is with the University of Rennes 1 in the Lagadic group at Irisa and Inria Rennes Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes Cedex, France riccardo.spica@irisa.fr

G. Claudio is with Inria in the Lagadic group at Irisa and Inria Rennes Bretagne Atlantique and Irisa, Campus de Beaulieu, 35042 Rennes Cedex, France giovanni.claudio@inria.fr

F. Spindler is with Inria in the Lagadic group at Irisa and Inria Rennes Bretagne Atlantique and Irisa, Campus de Beaulieu, 35042 Rennes Cedex, France fabien.spindler@inria.fr

P. Robuffo Giordano is with CNRS in the Lagadic group at Irisa and Inria Rennes Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes Cedex, France prg@irisa.fr

¹<http://coppeliarobotics.com/>

²<http://www.ros.org/>

³<http://www.irisa.fr/lagadic/visp/visp.html>

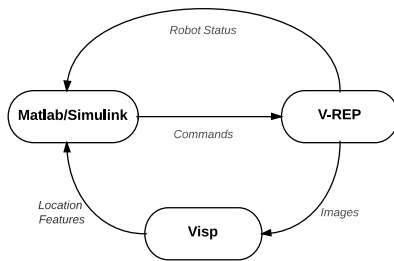


Fig. 2: Software modules and exchanged data.

to run at the same rate w.r.t. the real time (thus, one can also run faster than real-time simulations if needed). Finally, thanks to the ROS communication interface, it is also straightforward to plug the matlab/simulink controller with alternative simulation softwares or real hardware counterparts of the simulated robots (as long as the communication interface is the same). This is for instance the case for the simulated manipulator which has a real counterpart in our lab⁴.

To conclude, this presentation will show how to easily interface matlab/simulink with a widespread, open-source, and state-of-the-art 3D robotic simulation (V-REP) using ROS as communication middleware for sensor-based control of popular robotic platforms. A *live demo* of the simulation/control architecture will also be shown during the workshop.

We believe this contribution fits well with the workshop topics as it involves the use of matlab/simulink for sensor processing (in our case camera camera images and IMU data), control law testing (visual servoing, flight control), possibility to use the same code with the simulated and real robots (owing to the ROS middleware), and finally its strong educational impact as the V-REP simulation engine (and ROS) can be freely accessed by academic users.

REFERENCES

- [1] O. Bourquardez, R. Mahony, N. Guenard, F. Chaumette, T. Hamel, and L. Eck, "Image-based visual servo control of the translation kinematics of a quadrotor aerial vehicle," *IEEE Trans. on Robotics*, vol. 25, no. 3, pp. 743–749, 2009.
- [2] F. Chaumette and S. Hutchinson, "Visual servo control, Part I: Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.
- [3] E. Marchand, F. Spindler, and F. Chaumette, "ViSP for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robotics and Automation Magazine*, vol. 12, no. 4, pp. 40–52, 2005.

⁴<http://www.irisa.fr/lagadic/material/material-eng.html>